AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects, the method comprising the steps of:

changing an object in the first system;

determining an ~~minimal~~ object change set which ~~information~~ represents~~ing~~ ~~a~~ changes made to the object in the first system, wherein the object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object; and

~~distributing~~ sending the object change ~~information~~ set directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change ~~information~~ set ~~in~~to the corresponding object in the second system's cache so as to synchronize the second system with the first system, wherein the second system is registered in the first system prior to ~~distributing~~ sending the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

2. (Original) The method as claimed in claim 1 further comprising a step of establishing a communication link between the first system and the second system wherein the distributing step distributes the object change information from the first system to the second system through the communication link.

3. (Original) The method as claimed in claim 2 wherein the establishing step establishes the communication link based on a publish/subscribe protocol.

4. (Canceled).

5. (Original) The method as claimed in claim 1 further comprising a step of sending the object change information to a database for updating the object in the database with the object change information.

6. (Original) The method as claimed in claim 5 further comprising the steps of:

receiving an error message from the database when the updating of the object in the database fails; and

discarding the object change information prior to the distributing step in response to the error message.

7. (Original) The method as claimed in claim 1 wherein the first system includes an object cache for storing one or more objects, and the method further

comprises a step of merging the object change information into the object cache of the first system.

8. (Canceled).

9. (Canceled).

10. (Canceled).

11. (Original) The method as claimed in claim 1 wherein the first system includes a cache for storing one or more objects, the method further comprising the steps of:

receiving object change information distributed from the second system and containing information of changes made to one or more objects in the second system; and

merging the object change information received from the second system into the objects in the cache of the first system to synchronize the first system with the second system.

12. (Currently amended) A method for providing object change information from a first system to a second system for synchronizing the second system with the first system, the first system having a first object cache for storing one or more objects and the second system having a second object cache for storing one or more objects, the method comprising the steps of:

determining ~~minimal~~ object change sets ~~information~~ which represent~~ing~~ ~~a~~ changes made to ~~an~~ objects in the first system, wherein an object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contains the attribute names and the new attribute values of attributes that were changed in the object; and

~~distributing~~ sending the object change ~~information~~ sets directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change ~~information~~ sets ~~in~~to corresponding objects in the second object cache so as to synchronize the objects in the second cache of the second system with the changed objects in the first system, wherein the second system is registered in the first system prior to ~~distributing~~ sending the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

13. (Original) The method as claimed in claim 12 further comprising a step of establishing a communication link between the first system and the second system wherein the distributing step distributes the object change information from the first system to the second system through the communication link.

14. (Canceled).

15. (Original) The method as claimed in claim 12 further comprising a step of sending the object change information from the first system to a database for updating the object in the database with the object change information.

16. (Original) The method as claimed in claim 15 further comprising the steps of:

receiving an error message from the database when the updating of the object in the database fails; and

discarding the object change information prior to the distributing step in response to the error message.

17. (Canceled).

18. (Canceled).

19. (Canceled).

20. (Original) The method as claimed in claim 12 further comprising steps of:

receiving object change information distributed from the second system and containing information of changes made to one or more objects in the second system; and

merging the object change information received from the second system into the objects in the first cache of the first system to synchronize the first system with the second system.

21. (Currently amended) A synchronization executor for providing object change information from a first system to a second system for synchronizing the second system with the first system, the first system being capable of changing the

object, the second system having an object cache for storing objects, the system comprising;

a synchronization manager for obtaining ~~minimal~~ an object change ~~information~~ set which ~~representing~~ represents ~~a~~ changes made to an object in the first system, wherein the object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object; and

a dispatcher for distributing the object change ~~information~~ set directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change ~~information~~ set ~~in~~to the corresponding object in the second system's cache so as to synchronize the object in the second system with the first system, wherein the second system is registered in the first system prior to distributing the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

22. (Original) The executor as claimed in claim 21 wherein the synchronization manager establishes a communication link with the second system and the dispatcher distributes the object change information to the second system through the established communication link.

23. (Previously presented) The executor as claimed in claim 22 wherein the synchronization manager establishes the communication link based on a publish/subscribe protocol.

24. (Original) The executor as claimed in claim 21 further comprising a connector for obtaining the object change information that is distributed from the second system.

25. (Canceled).

26. (Canceled).

27. (Canceled).

28. (Currently amended) A persistence system for synchronizing an object on a network, the network including a caching system having an object cache for storing objects, the persistence system comprising;

a transaction manager for changing an object and determining an ~~minimal~~ object change set which ~~information~~ represents~~ing~~ ~~the~~ changes made to the object, wherein the object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object ~~for updating a database~~; and

a synchronization executor for obtaining the object change ~~information~~ set from the transaction manager and distributing the object change ~~information~~ set to the caching system to cause the caching system to ~~merge~~ apply the object change ~~information~~set ~~in~~to the corresponding object in the cache so as to synchronize the object in the object cache with the changed object in the persistence system, wherein the second system is registered in the first system prior to distributing the object change information directly from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

29. (Original) The system as claimed in claim 28 further comprising a persistence system cache for storing one or more objects.

30. (Canceled).

31. (Original) The system as claimed in claim 28 wherein the synchronization executor establishes the network, and the dispatcher distributes the object change information via the established network.

32. (Currently amended) Computer readable media storing instructions for use in the execution in a computer of a method for providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects, the method comprising the steps of:

changing an object in the first system;

determining ~~minimal~~ an object change ~~information~~ set which ~~representing~~ represents changes made to the object in the first system, wherein the object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object; and

~~distributing~~ sending the object change ~~information~~ set directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change ~~information~~ set ~~in~~to the corresponding object in the second system's cache so as to synchronize the second system with the first system, wherein the second system is registered in the first system prior to ~~distributing~~ sending the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

33. (Currently amended) Electric signals for execution in a computer of a method for providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects, the method comprising the steps of:

changing an object in the first system;

determining ~~minimal~~ an object change ~~information~~ set which ~~representing~~ represents changes made to the object in the first system, wherein the object change set includes:

a primary key value that identifies the object; and

a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object; and

~~distributing~~ sending the object change ~~information~~ set directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change ~~information~~ set ~~in~~to the corresponding object in the second system's cache so as to synchronize the second system with the first system, wherein the second system is registered in the first system prior to ~~distributing~~ sending the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.

34. (Currently amended) A computer program product for execution in a computer of a method for providing object change information from a first system to a second system for synchronizing the second system with the first system, the second system having an object cache for storing objects, the computer program product comprising:

a module for changing an object in the first system;

a module for determining an ~~minimal~~ object change ~~information~~ set which ~~representing~~ represents ~~a~~ changes made to the object in the first system, wherein an object change set includes:

a primary key value that identifies the object; and a set of attribute changes which contain the attribute names and the new attribute values of attributes that were changed in the object; and

a module for ~~distributing~~ sending the object change ~~information~~ set directly from the first system to the second system to cause the second system to ~~merge~~ apply the object change information ~~in~~to the corresponding object in the second system's cache so as to synchronize the second system with the first system, wherein the second system is registered in the first system prior to distributing the object change information from the first system to the second system, wherein the registration process causes the first system to know that the second system wants to receive object change information of objects changed in the first system, thereby eliminating the need for the second system to register with every object it is interested in.